



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/649,198	08/27/2003	Panagiotis Kougiouris	11389-033	5126
20583	7590	05/22/2006	EXAMINER	
JONES DAY 222 EAST 41ST ST NEW YORK, NY 10017			NGUYEN, MAIKHANH	
			ART UNIT	PAPER NUMBER
			2176	

DATE MAILED: 05/22/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/649,198	Applicant(s) KOUGIOURIS ET AL.	
	Examiner Maikhanh Nguyen	Art Unit 2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 23 February 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-19 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-19 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to communications: Amendment filed 02/23/2006 to the original application filed on 08/27/2003, which is Continuation of No. 09/441188, filed 11/15/1999.
2. Claims 1-19 are currently pending in this application. Claims 1-12, 16-17, and 19 have been amended. Claim 1 is independent claim.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

(b) This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1-15 and 17-19 maintain rejected under 35 U.S.C. 103(a) as being unpatentable over **Ferris et al.** (U.S. 6,253,228, filed 03/1997) in view of **Strong** (U.S. 6,167,523, filed 05/1997).

Ferris and Strong references were cited in the Applicant's IDS, filed 08/27/2003.

As to claim 1:

Ferris teaches a method for automatically validating text input, the method comprising:

- processing a markup language file (*e.g., a Web page*) comprising a description of a graphical user interface, the description comprising a GUI element enable to receive text input (*e.g., an interface involves two text fields ... The first field is for input ... The user enters a value in the input field*) [col.9, lines 57-67 & col. 10, lines 35-37], wherein the markup language file comprises a markup language tag for instantiating a validation manager (*e.g., instances of the object classes are instantiated for the HTML elements defined in the HTML document ... instance variables of an object and provide runtime environment information such as object, method and variable binding; col.7, line 25-col.8, line 59*);
- instantiating the validation manager in response to said processing the markup language file (*e.g., The name attribute of the WEBOBJECT tag binds a WEBOBJECT HTML template entry to the declarations file. For example, the name having a value of "INPUTFIELD" (from the WEBOBJECT tag of Table 1) binds its entry to the INPUTFIELD entry in the declarations file. A similar approach can be taken for the BUTTON and OUTPUTFIELD values associated with the remaining WEBOBJECT tags. The INPUTFIELD, BUTTON, and OUTPUTFIELD declarations bind the WEBOBJECT tag to instances of the WOApplet class. The WOApplet class permits the specification of applet-specific*

parameters, such as the dimensions of the applet and the location of the ".class" file to download to the browser. It also allows you to initialize parameters to be downloaded to the applet and to bind an applet's keys to variables and methods in the server; col.8, lines 29-59);

- displaying the GUI on a display screen of a computer system in response to said processing the markup language file (*e.g., rerender the applet on the client's display ... by the browser; col.10, lines 1-10);*
- receiving text input to the GUI element enabled to receive text input [*e.g., a browser user enters information in an input field of the Web page ... numerous applets that are capable of data display and/or input ... various applets are provided to the user including: a TextFieldApplet (a text field permitting text input from a user); a ButtonApplet (a push button); a CheckboxApplet (a toggle button); a ChoiceApplet (a box containing non-editable items that allows multiple selection); a ListApplet (a list of items that can allow multiple selection); a RadioGroupApplet (a set of mutually exclusive buttons); a ScrollingTextApplet (a larger text entry area); CalendarApplet (a primitive graphical calendar for entering or displaying dates); and a StatusTextApplet (displays a read-only message center aligned within the applet's frame). Each applet may contain multiple values (e.g. the text in a TextFieldApplet or the selected item in the ListApplet); col.10, lines 36-63]; and*
- receiving a programmatic event by the validation manager in response to the received text input to the GUI element (*e.g., Upon receiving invocation*

instructions from the instance of Association 302, Action Coordinator 301 queries Associations 302 for the values for all of their keys at step 604. The values retrieved are examined to determine whether the value has been changed since the last communication with the server 'e.g., the values retrieved are compared to the values in the dictionary'. If any values have changed, synchronization with the server takes place ... The server invokes the action at step 610. Subsequent to the invocation of the action, the server constructs a package at step 611 to be sent back to the browser. In constructing the package, the server determines whether a new HTML document/page is to be rendered. If a new page is to be rendered, an HTML document is created and the URL for that document is transmitted back to the Applet Group Controller 202 in the browser. The browser then retrieves the HTML document from the URL. However, if the state merely needs to be synchronized on the browser, there is no need to generate a new page. Instead, the bindings are evaluated and the states are packaged together; see the discussion, beginning at col.13, line 60).

Ferris, however, does not specifically teach “*determining whether the text input received to the GUI element is valid text input in response to receiving the programmatic event; and providing an indication that the text input received to the GUI element is invalid if the text input is determined to be invalid.*”

Strong teaches:

- determining whether the text input received to the GUI element is valid text input in response to receiving the programmatic event (*e.g., If no invalid input data is identified, one or more data processing programs, are also referred to herein as "handlers," are invoked by the data validation and processing control program to process the input data; col.3, lines 44-47*); and
- in providing an indication that the text input received to the GUI element is invalid if the text input is determined to be invalid (*e.g., if the data in the field is invalid according to requirements specified in the one or more configuration registry keys, an error message corresponding to the field being evaluated is dynamically built and logged in an error log; col.3, lines 36-40*).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Strong with Ferris because it would have provided the capability for controlling processing of input data from electronic forms such as HTML forms.

As to claim 2:

Ferris teaches an attribute for specifying a type associated with the GUI element, the method further comprising: the validation manager instantiating a validation component, wherein said instantiating a validation component comprises specifying the type associated with the GUI element (*e.g., determines the appropriate action logic to invoke; Fig.6A & col.11, line 50-col. 12, line 23*); wherein said validation manager determining

whether the text input at the GUI element is valid text input comprises the validation manager calling the validation component (*e.g., invoke action; Fig.6A*); wherein said validation manager calling the validation component comprises the validation manager specifying the text input to the GUI element; wherein the validation is operable to return a result value to the validation manager indicating whether the text input received to the GUI element is valid text for the type associated with the GUI element (*e.g., transmit package to Applet Group Controller; Fig.6A*).

As to claim 3:

Ferris teaches receiving text input to the GUI element is performed by a user of the application (*e.g., an interface involves two fields ... for input ... The user enter a value in the input field; col.9, lines 59-67 & col. 10, 48-67*).

As to claim 4:

Ferris teaches the step of received text input to the GUI element comprises an application providing text input to the GUI element (*e.g., various applets are provided to the user ... a text field permitting text input from a user; col.10, lines 48-67*).

As to claim 5:

Strong discloses one or more attributes for specifying when text input to the GUI element should be validated; the validation manager component is operable to validate text input to the GUI element in accordance with the one or more attributes specifying when text

input to the GUI element should be validated (*e.g., an HTML form that includes one or more fields for data input ... then the form can be validated by the forms data validation; col.3, lines 24-47*).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Strong with Ferris because it would have provided the capability for determining the validate data input from user.

As to claim 6:

Ferris teaches each of the one or more attributes for specifying when text input to the GUI element should be validated corresponds to at least one type of programmatic event (*see fig.6B and the associated text*); the step of said validation manager receiving a programmatic event comprises the validation manager ignoring the programmatic event if the programmatic event does not correspond to one of the attributes for specifying when text input to the GUI element should be validated (*col.13, lines 49-58*).

As to claim 7:

Ferris teaches receiving one or more the following actions: clicking on the GUI element (*see the clicking of a button icon discussion beginning at col.13, line 49*); wherein said validation manager component receiving a programmatic event in response to said providing text input to the GUI element comprises the validation manager component receiving a programmatic event corresponding to the action performed (*Figs. 3 and 6A,*

in a preferred embodiment, upon the occurrence of an event ... the clicking of a button icon ...invoke the action; col.13, lines 49-59).

As to claim 8:

Ferris teaches one or more parameters (*e.g., Bind variables to keys and values; Fig.6A*) for specifying the default behavior of when the validation manager should validate text input for GUI elements described in the markup language file (*col.8, lines 29-59*).

As to claim 9:

Ferris teaches one or more attributes for specifying when text input to the GUI element should be validated (*see fig.6B &the associated text & col.11, line 50-col. 12, line 23*); the validation manager is operable to override the default behavior and validate text input to the GUI element in accordance with the one or more attributes specifying when text input **received** to the GUI element should be validated (*e.g., transmit changed values to association, filter changed values; Fig.6B*).

As to claim 10:

Ferris teaches said validation manager indicating that the text input to received to the GUI element is invalid comprises the validation manager requesting the application to alter the visual appearance of the GUI element (*e.g., any state that has changed in the client... anything that action has caused to be changed, is reflected to the client for the user's viewing; col.10, lines 34-47*).

As to claim 11:

Ferris teaches validation manager indicating that the text input received to the GUI element is invalid comprises the validation manager displaying an informational user interface window (*e.g., any state that has changed in the client... anything that action has caused to be changed, is reflected to the client for the user's viewing; col.10, lines 34-47*).

As to claim 12:

Ferris teaches one or more attributes for controlling text input validation for the GUI element (*col. 12, line 55-col.13, line 11*); wherein said step of processing a markup language file comprises the application constructing a document object representing the markup language file; wherein instantiating the validation manager comprises the application passing a reference to the document object to the validation manager (*col.7, line 25-col.8, line 59*); wherein, in response to being instantiated by and receiving the reference to the document object, the validation manager is operable to traverse the document object in order to discover the one or more attributes for controlling text input validation for the GUI element (*col.4, lines 54-col.5, line 9*).

As to claim 13:

Ferris teaches the one or more attributes for controlling text input validation for the GUI element include an attribute for specifying a type associated with the GUI element (*e.g., a string capitalizer; col.9, lines 51-60*).

As to claim 14:

Ferris teaches the one or more attributes for controlling text input validation for the GUI element includes one or more attributes for specifying when text input to the GUI element should be validated (*e.g., determines the appropriate action logic to invoke; Fig.6A*).

As to claim 15:

Strong teaches the one or more attributes for controlling text input validation for the GUI element include one or more attributes for specifying how invalid text input for the GUI element should be indicated (*e.g., if the data in the field is invalid according to requirements specified in the one or more configuration registry keys, an error message corresponding to the field being evaluated is dynamically built and logged in an error log; col.3, lines 36-40*).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Strong with Ferris because it would have provided the capability for handling validation of input data.

As to claim 17:

Ferris teaches the validation manager component is a Java object (*e.g., The applets can be written using a programming language such as Java; col.6, lines 26-43*).

As to claim 18:

Ferris teaches the markup language is HTML (*e.g., HTML; col.4, lines 41-53 & col.6, line 66- col.7, line 8*).

As to claim 19:

Ferris teaches the type associated with the GUI element is a type comprising, among other things, U.S. states and territories (*e.g., the information collected on a form ...CITY/HOUSTON; col.3, lines 61-67*).

4. Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Ferris et al.** in view of **Strong** as applied to claim 1 above, and further in view of **Microsoft Corporation**, "Microsoft Component Services: Server Operating System A Technology Overview" (last updated August, 15 1998).

As to claim 16:

The combination of Ferris and Strong do not specifically teach “the validation manager component is a COM object.”

Microsoft teaches the validation manager component is a COM object (*e.g., COM makes it easy to develop powerful component-base application; page 2*).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Microsoft with Ferris as modified by Strong because it would have provided the capability for implementing an efficient means to validate data in an information processing system.

Response to Arguments

5. Applicant’s arguments filed 02/23/2006 have been fully considered but they are not persuasive.
 - a. Applicant argues that Ferris does not disclose receiving text input to a GUI element enabled to receive text input displayed as a result of processing a markup language file comprising the description of that GUI element [Remarks, page 8].

In response, Ferris does teach receiving text input to a GUI element enabled to receive text input displayed as a result of processing a markup language file

comprising the description of that GUI element (*e.g., an interface involves two text fields ... The first field is for input ... The user enters a value in the input field*) [col.9, lines 57-67 & col. 10, lines 35-37].

- b. Applicant argues that Ferris does not disclose instantiating the validation manger in response to the processing the markup language file [Remarks, page 9, 2nd ¶].

In response, Ferris teaches instantiating the validation manger in response to the processing the markup language file (*e.g., applets that run on a client ...a Web page and can be programmed to produced an unlimited number of input mechanisms to a user; col.4, lines 47-53 & a programming language such as Java include: the ability to write an unlimited number and variety of programs, Java applications will run on almost any supporting platform without the need to recompile the code, Java is widely used in WWW applications, and a Java program may be embedded into a Web page ... a Java program may execute its action logic on either a client or a server ...the applet is treated as though it were an HTML element with an applet tag. The applet tag specifies the location of the class file 'a class file contains a description of a specific kind of programmed object, including the instructions that are particular to that object' and the display area allocated to the applet; col.6, lines 30-65).*

- c. Applicant argues that Ferris does not teach using HTML form [Remarks, page 9].

In response, Applicant is arguing the disclosure, not the claimed limitations.

“HTML forms” is not claimed. However, Ferris does teach HTML forms

[e.g., a browser user enters information in an input field of the Web page ...

numerous applets that are capable of data display and/or input ... various applets

are provided to the user including: a TextFieldApplet (a text field permitting text

input from a user); a ButtonApplet (a push button); a CheckboxApplet (a toggle

button); a ChoiceApplet (a box containing non-editable items that allows multiple

selection); a ListApplet (a list of items that can allow multiple selection); a

RadioGroupApplet (a set of mutually exclusive buttons); a ScrollingTextApplet (a

larger text entry area); CalendarApplet (a primitive graphical calendar for

entering or displaying dates); and a StatusTextApplet (displays a read-only

message center aligned within the applet's frame). Each applet may contain

multiple values (e.g. the text in a TextFieldApplet or the selected item in the

ListApplet); col.10, lines 36-63].

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Jellinek et al.	U.S. Pat. No. 5,736,984	Issued: Apr. 7, 1998
-----------------	-------------------------	----------------------

Limsico	U.S. Pat. No. 5,793,952	Issued: Aug. 11, 1998
---------	-------------------------	-----------------------

Bahrs et al.

U.S. Pat. No. 6,654,932

Issued: Nov. 25, 2003

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a). A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

Contact Information

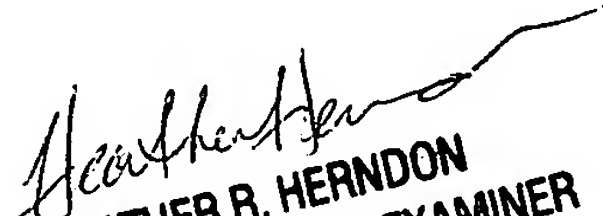
8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Maikhanh Nguyen whose telephone number is (571) 272-4093. The examiner can normally be reached on Monday - Friday from 9:00am – 5:30 pm. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather Herndon can be reached on (571) 272-4136.

Art Unit: 2176

The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MN


HEATHER R. HERNDON
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100